Observation of the Evolution of Hide and Seek AI

CSC 492 Senior Project II
Completion Date: June 2021

California Polytechnic University, San Luis Obispo
Computer Science and Software Engineering Department

Author: Anthony Catelani
Advisor: Professor Franz J. Kurfess

Abstract

The purpose of this project is to observe the evolution of two artificial agents, a 'Seeker' and a 'Hider', as they play a simplified version of the game Hide and Seek. These agents will improve through machine learning, and will only be given an understanding of the rules of the game and the ability to navigate through the grid-like space where the game shall be played; they will not be taught or given any strategies, and will be made to learn from a clean slate. Of particular interest is observing the particular playstyle of hider and seeker intelligences as new elements are introduced into the game, such as obstacles, doors, among other environmental influences. Through this observation, I hope to identify not only key strategies in the game of hide and seek, but to achieve a greater understanding of the evolution of machine learning AI searching and hiding patterns, which are relevant to several fields such as networking, artificial intelligence, and cyber security.

Introduction

In this project, I hope to establish a greater understanding of the nature of how seeker intelligences learn over the course of several generations to better play a game of Hide and Seek, and observe the various strategies and development patterns that arise over the course of the process. To this end, I have created a computer simulation of the game Hide and Seek, and a process by which the agent can learn over the course of several thousand games. Intuitively speaking, I would expect that as the seeker intelligence learns, it will approach a near certain percentage chance of victory, as if the agent learns to properly trace the path left behind by the hider, then it should in theory always find them in time, though due to issues in my current implementation, it seems my seeker intelligence only ever becomes marginally better than true random.

Background

The game of hide and seek has been around for a long time; the oldest variant that historians are aware of dates back to the second century B.C. in Greece, where it was named *apodidraskinda*. The modern version of the game has two types of players, a hider and a seeker. At the start of the game, the hiders are given a set amount of time to hide in a given area, which the seeker typically keeps track of. Once time is up, the seeker begins to search for the hiders. The seeker wins once all hiders have been found, at which point the game either repeats or ends.

Related Work

The most prominent example of computer intelligences using the game of hide and seek as a learning tool was a study on emergent tool use [1], where AI learned to pick up blocks and platforms in order to properly hide themselves from seekers. There is also another study where the game of hide and seek was used to analyze computer networking systems [3].

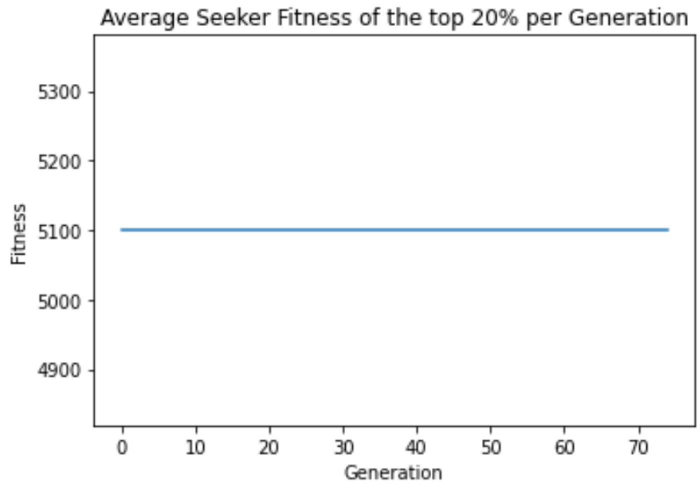
Implementation

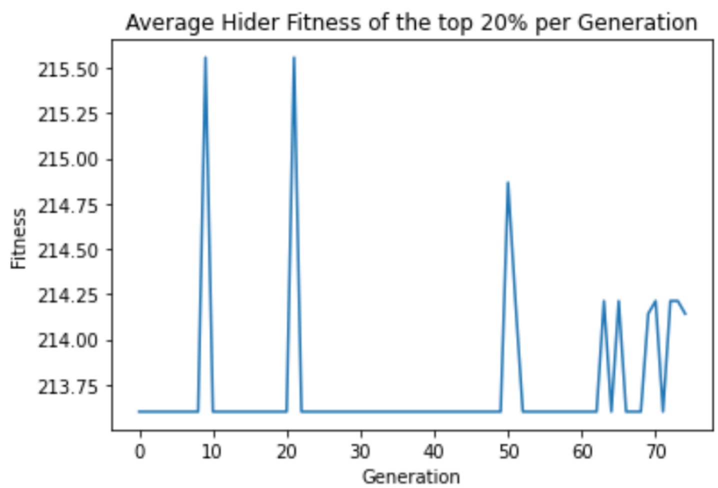
My hide and seek simulation is coded in Python, as its nature as an object-oriented language makes machine learning that much easier to work with, and hosted on a Jupyter Notebook, for ease of use. Using Numpy, I constructed a 20 by 20 grid, hereafter referred to as the playground, wherein the simulated game of hide and seek takes place. To facilitate machine learning, I implemented a genetic algorithm, wherein seeker intelligences are tested and measured in order to create an optimal seeker intelligence over the course of several games and several generations. During development, there were several issues plaguing the results of the evolution, which I had attributed to stupid hiders rewarding stupid seekers. Newly instantiated hider intelligences would typically hide right near the origin, which was a problem considering that meant that my fitness function was greatly rewarding any seeker that played against it, regardless of that seeker's actual quality. So I had decided to remove it after much deliberation. However, after much researching and rewriting a good chunk of my fitness functions, repopulation functions, instantiation functions, and the movement methods, I reconsidered the possibly of re-adding the hider intelligence, and decided that my new implementation could account for stupid hiders. Reimplementing the hider intelligence did cost me a lot of time that would've been put towards adding hazards, but I decided that adding in the hider would result in enough of a nuanced game for the purposes of observation. So at the end of the third phase of testing, the only environmental obstacle I had implemented were rocks that I scattered randomly across each playground that neither AI could move over. The hider is given a set amount of time (approximately 50 moves) to move around the playground and choose a hiding spot, after which the seeker will also be given the same amount of time to find the hider. If the hider can successfully evade the seeker, they win, otherwise the seeker wins. The seeker will also create a scent, which decays over time, yet points a path towards the hider, which was implemented in order to give the seeker a way to properly learn, rather than blindly guess the hider location.

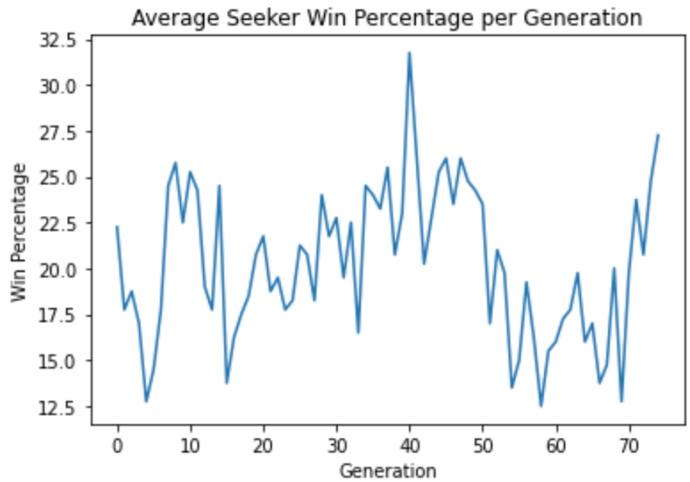
For a greater look at the code itself, it can be accessed through this GitHub link:
https://github.com/acatelan/HideAndSeek

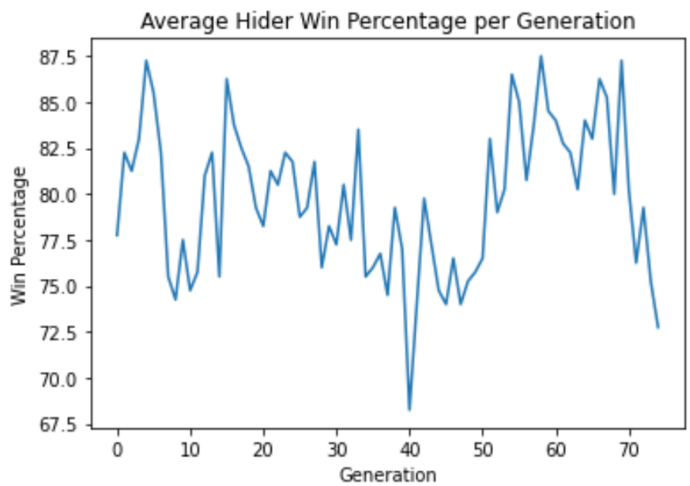
Testing and Validation

The first stage of testing occurred before the hider intelligence was removed. The hider and seeker had been given 75 generations and around 30,000 total games to learn; however, as is evident in the graphs, the two intelligences appeared to have not learned much at all, which I personally hold the hider intelligence to blame for.

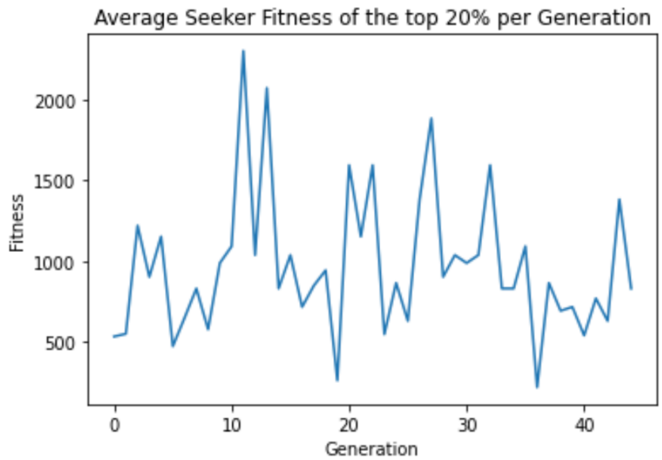

Average Seeker Fitness of the top 20% per Generation
Fitness
Generation
5300
5200
5100
5000
4900
0
10
20
30
40
50
60
70

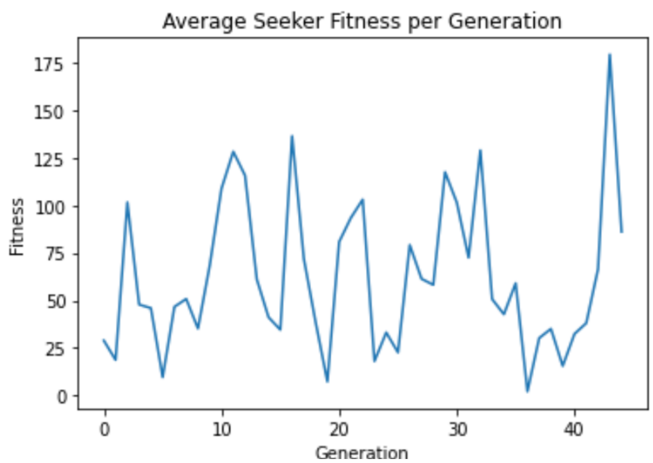

Average Hider Fitness of the top 20% per Generation
Fitness
Generation
215.50
215.25
215.00
214.75
214.50
214.25
214.00
213.75
0
10
20
30
40
50
60
70

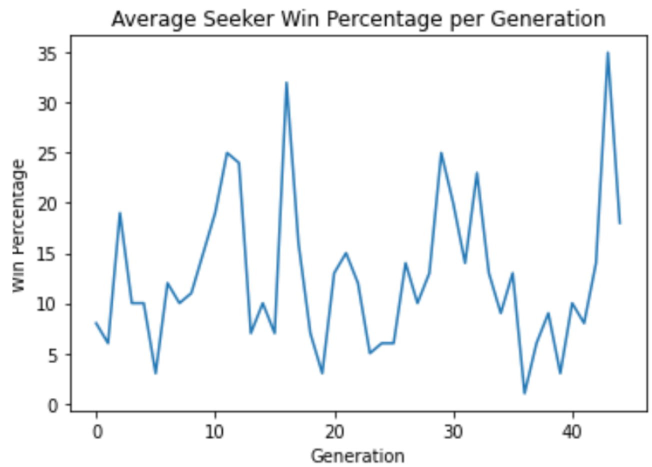


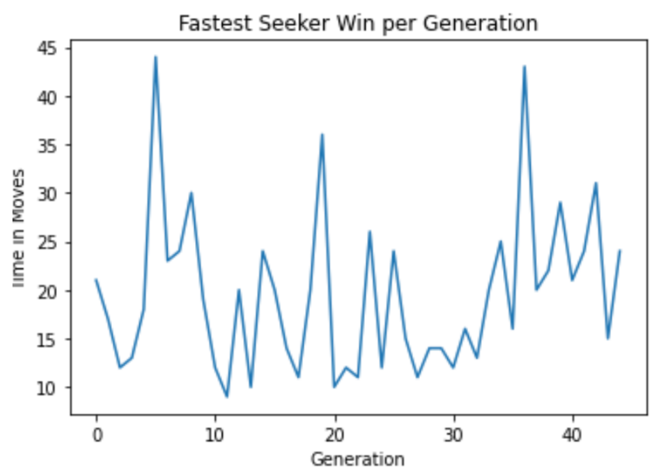


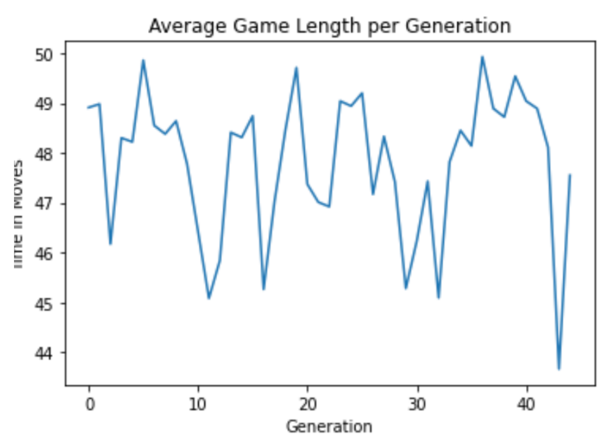
The second stage of testing began in a similarly empty playground, although this time the hider was removed. The genetic algorithm was also updated; now rather than simply randomly mutating the top 20% to get the newest generation, proper recombination occurs. Seeker AI now also starts with an additional array that they use to determine bias towards scents, which also evolves over the generations. The results of running 100 AI over the course of 45 generations can be seen in the charts below. As can be plainly seen, already we observe a dramatic change in fitness over time, which is no longer a flat line as a result of stupid hiders. We can also observe the beginning patterns of learning, as seen in a slow upwards trajectory of average seeker fitness.


Average Seeker Fitness of the top 20% per Generation
Fitness
2000
1500
1000
500
0
10
20
30
40
Generation


Average Seeker Fitness per Generation
Fitness
175
150
125
100
75
50
25
0
0
10
20
30
40
Generation



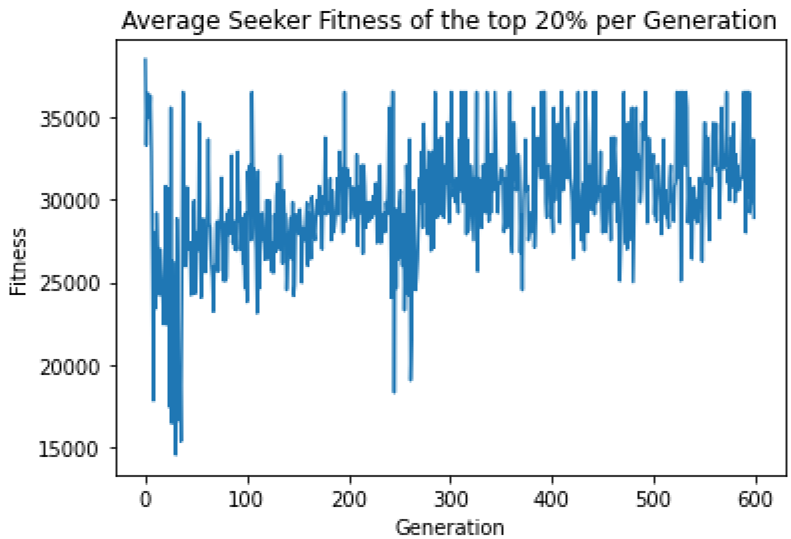


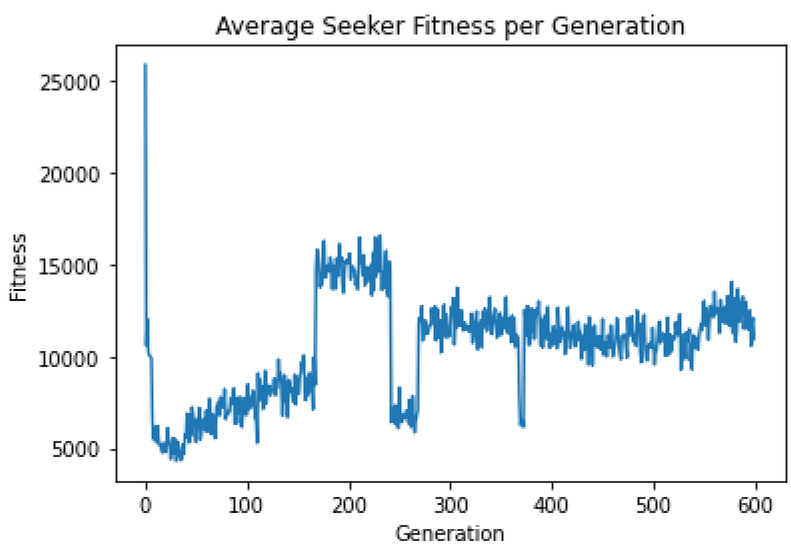


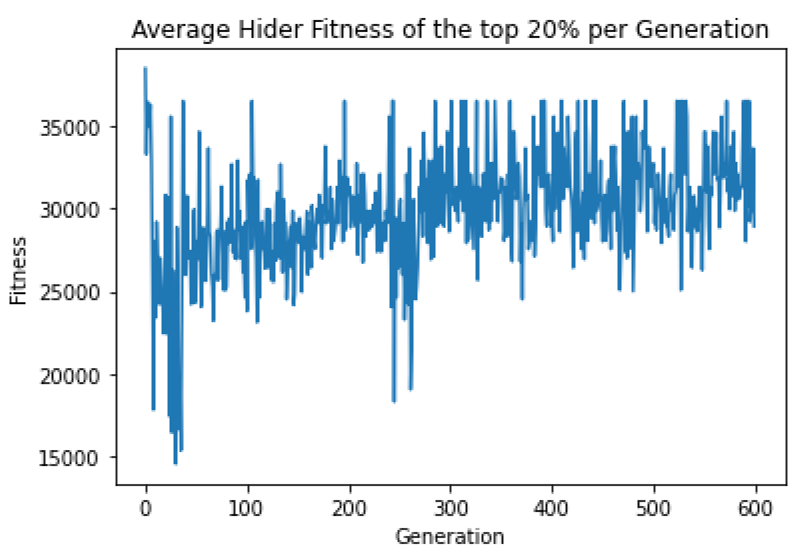
As I entered the third and final stage of development, I reconsidered the removal of the hider intelligence. The reason for this was due to my re-writing of the movement matrices. Originally, I had made a weird decision to create two matrices, one that contained all movement directions and one that contained all smell directions, and it would be these two matrices that would be re-combined and mutated over the course of the generations. However, I had thought that this implementation biased far more heavily in terms of direction rather than any actual sensory inputs. To amend this, I rewrote how my movement functions work. Now the intelligences evolve using an array of bias values. These values correspond to certain phenomena or actions that may impact an AI's movement decision. By the end of the third stage of testing, this array contained six different biases, which are, in order: smell, distance from the origin, active movement (moving around as opposed to standing still), previous movement (what was the last turn's movement matrix?), time elapsed, and passive movement (standing still as opposed to moving around). Those were the biases that I found most important.

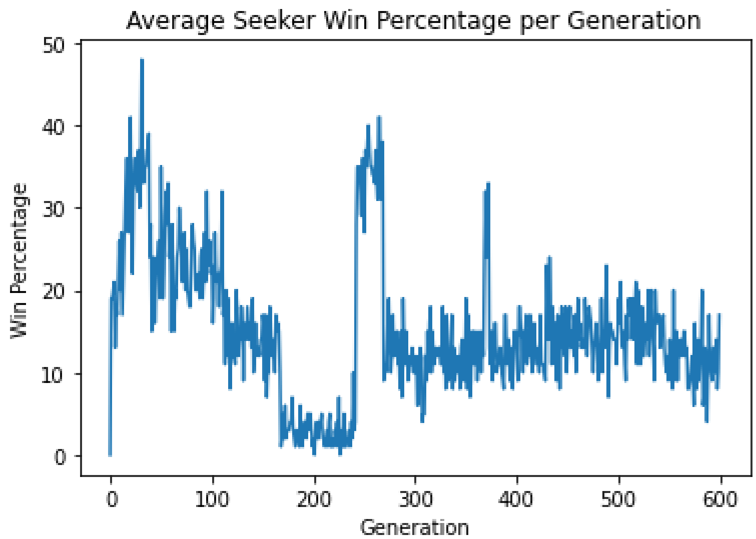
I set the program running for 600 generations with a population size of 100 seekers and 100 hiders and gathered the data. The first thing to note is that the hiders appear to have a great advantage. I had run this test multiple times, and would typically observe a seeker win rate of between 35-25%. However, it appears that in this instance of testing the hiders came up with some form of strategy that resulted in only around a 10% win rate for the seeker. It might be worth considering increasing the time allowed for the seeker to find the hider, but that is merely speculation at this point. What is very nice to see in the average seeker fitness graph is that there is a distinct upwards trend. I feel much more confident now that learning is indeed occurring, however slowly it may be. I also added graphs depicting the trends of the six bias values for both the hiders and seekers. It appears that my changing of the reproduction function resulted in more discrete evolution, rather than continuous, but I appreciate the clarity that it provides, so I am in no hurry to change it. One might initially be led into thinking that their evolution is random, which I had initially thought after looking them over, but then I took a look at the average bias for smell for the seeker. Intuitively, this is the most powerful tool in the seeker's arsenal, as it indicates a direct path straight towards the hider, following along their path, so it would be expected that if the seekers are indeed learning that they would value this trait highly. And as can be clearly observed, they do indeed value it highly, reaching the maximum value over the course of around 20-30 generations, and maintaining that value for the rest of the test. This leads me to believe that the seemingly random changes in the other biases are a result of strategy changes; reactionary evolutions in response to changes in the other AI's strategy.

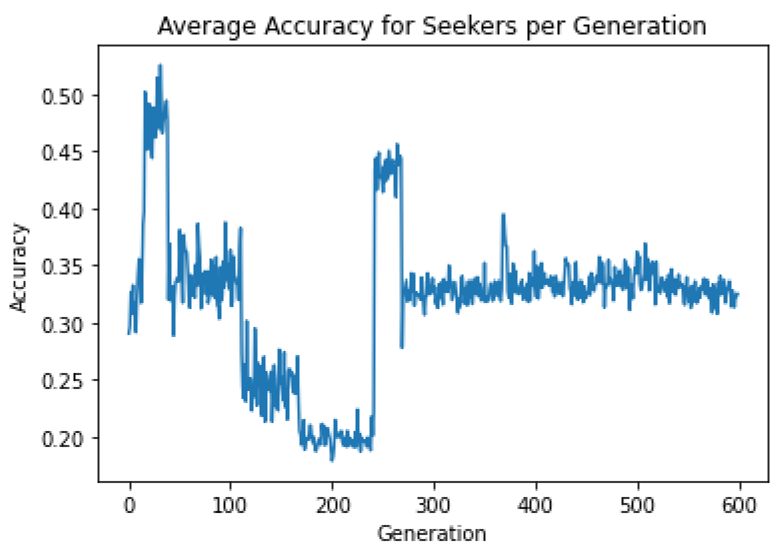
I also added an accuracy value; this value is updated each time the seeker makes a move. Accuracy is determined by taking the hider's position and the seeker's next movement position, and calculating the angle between the two. The greater the difference, the greater the inaccuracy. Though it is in hindsight that this might not be the best measurement, considering that hiders can create a wild goose chase, such that by following the smell, the seekers are moving away from the hider position, resulting in a lower accuracy rating and thus a lower fitness rating. Perhaps this is the strategy the hiders had created.

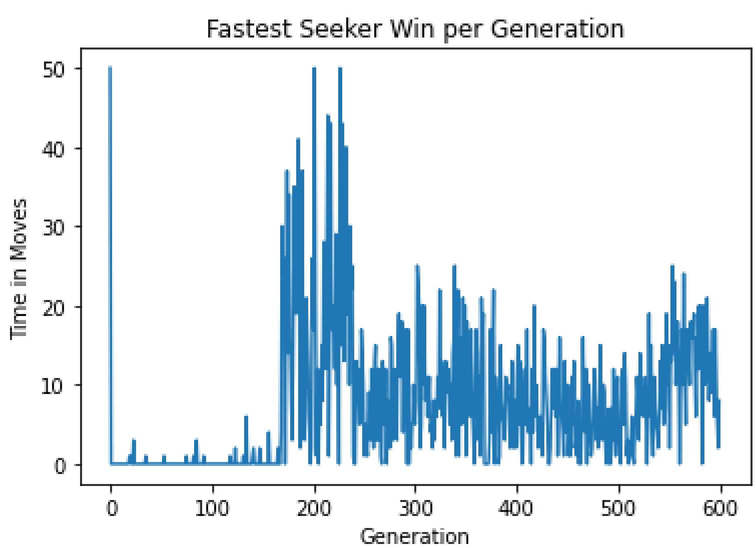

Average Seeker Fitness of the top 20% per Generation
Fitness
Generation
35000
30000
25000
20000
15000
0
100
200
300
400
500
600

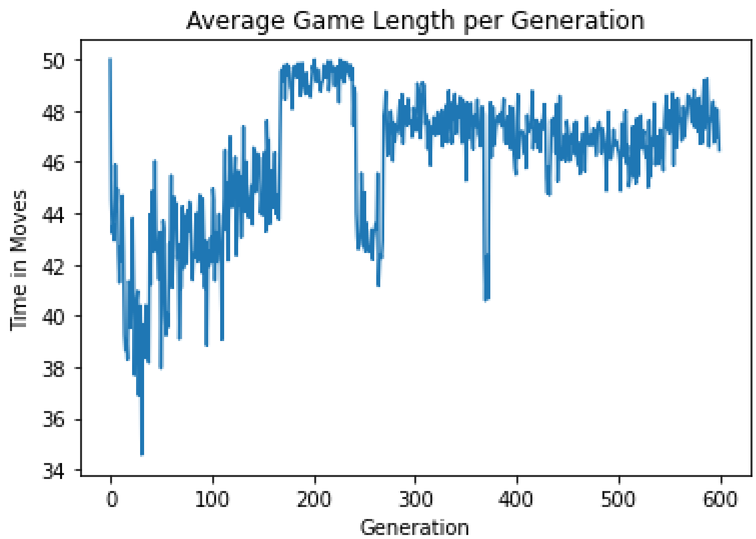

Average Seeker Fitness per Generation
Fitness
Generation
25000
20000
15000
10000
5000
0
100
200
300
400
500
600

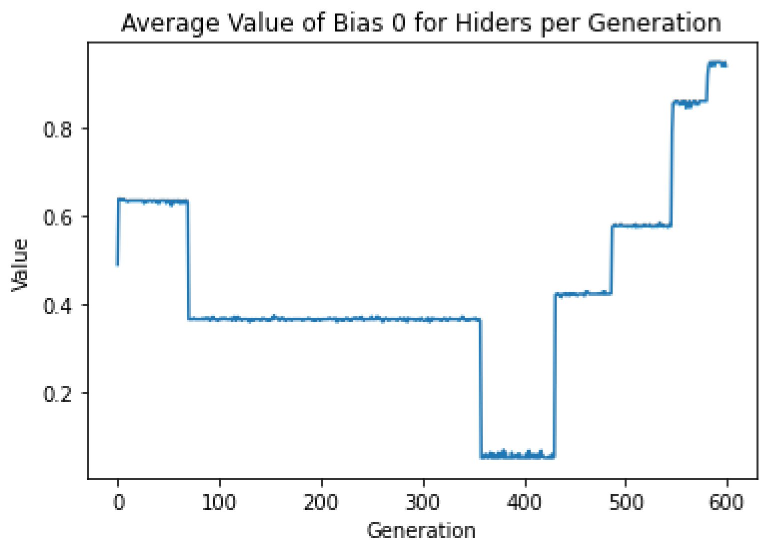

Average Hider Fitness of the top 20% per Generation
Fitness
Generation
35000
30000
25000
20000
15000
0
100
200
300
400
500
600


Average Seeker Win Percentage per Generation
Win Percentage
0
10
20
30
40
50
0
100
200
300
400
500
600
Generation

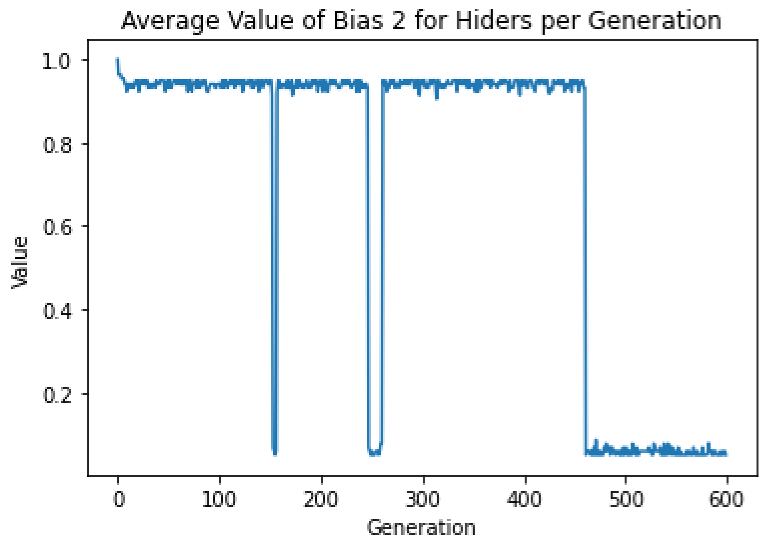

Average Accuracy for Seekers per Generation
Accuracy
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0
100
200
300
400
500
600
Generation

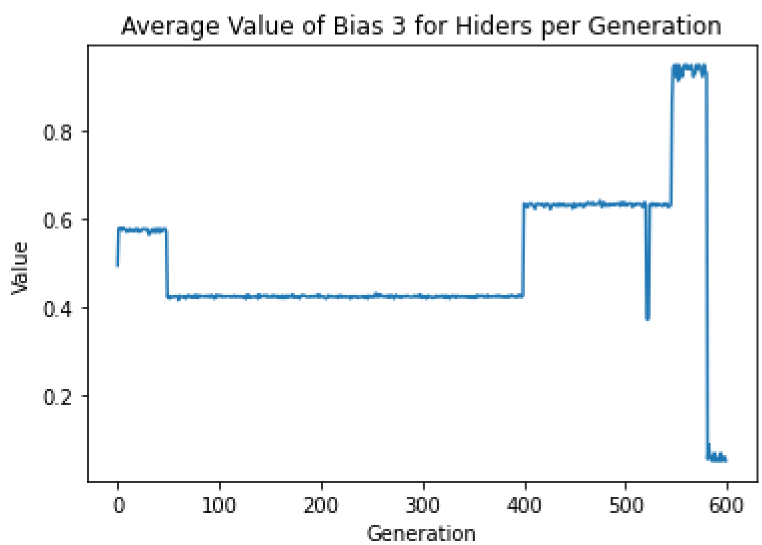

Fastest Seeker Win per Generation
Time in Moves
0
10
20
30
40
50
0
100
200
300
400
500
600
Generation

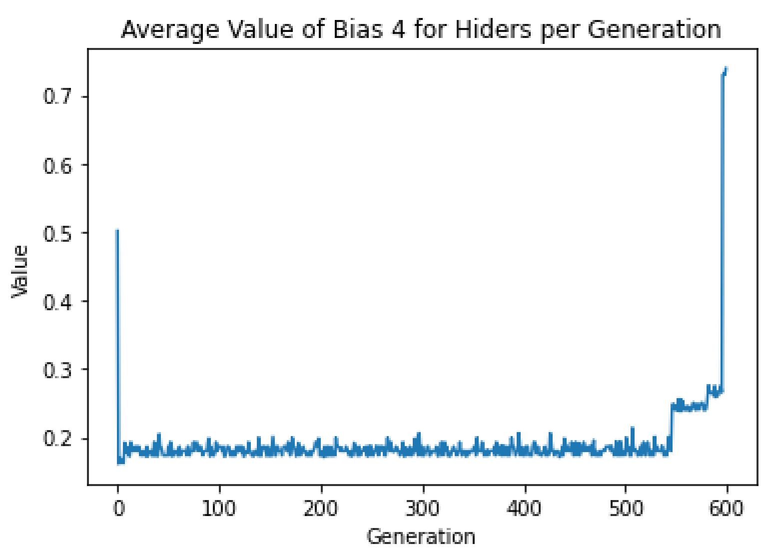

Average Game Length per Generation
Time in Moves
50
48
46
44
42
40
38
36
34
0
100
200
300
400
500
600
Generation

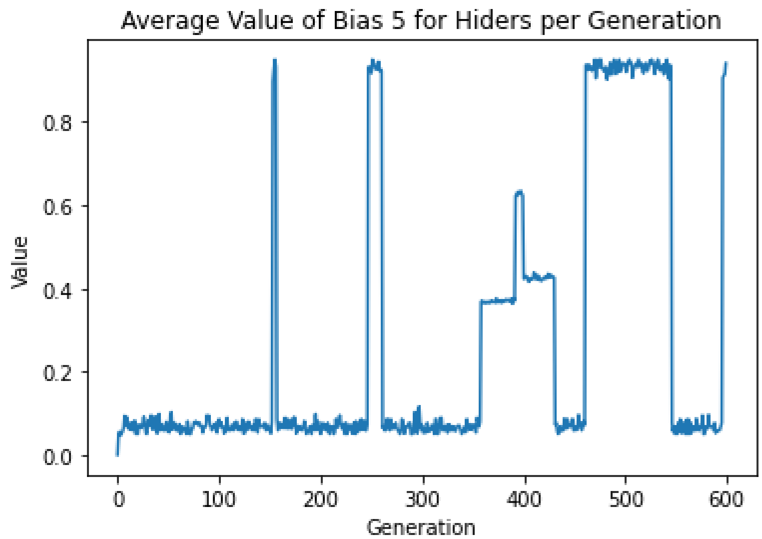

Average Value of Bias 0 for Hiders per Generation
Value
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600
Generation

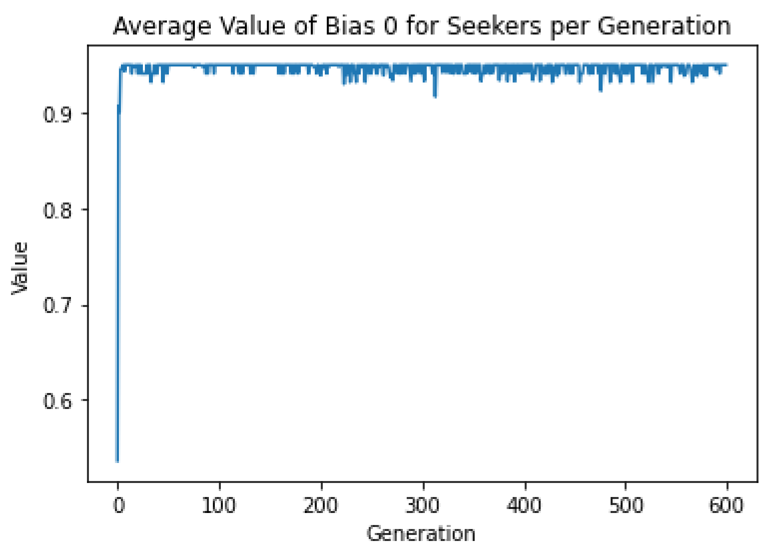

Average Value of Bias 1 for Hiders per Generation
Value
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600
Generation

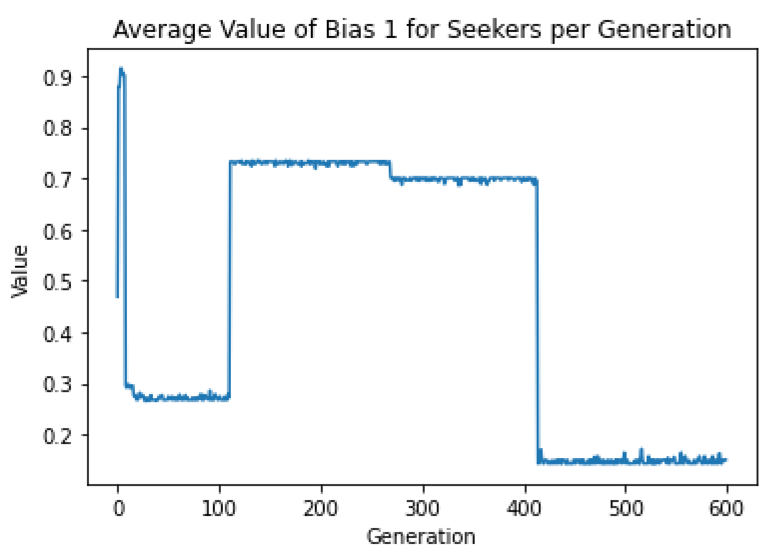

Average Value of Bias 2 for Hiders per Generation
Value
Generation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600


Average Value of Bias 3 for Hiders per Generation
Value
Generation
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600

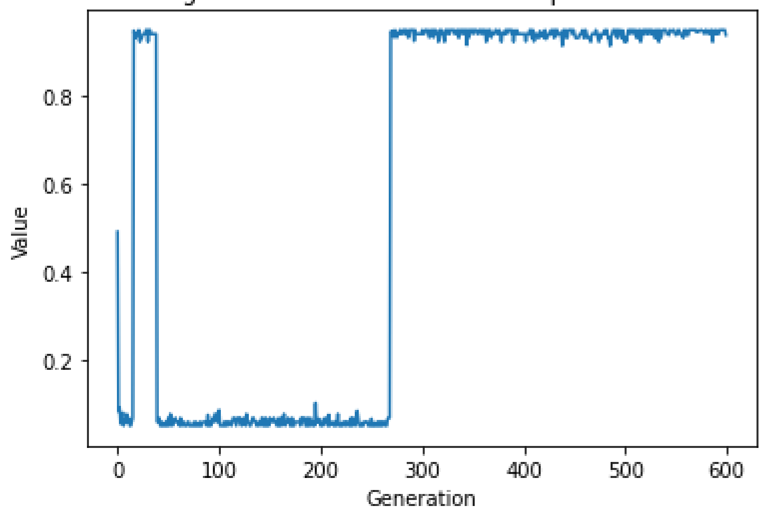

Average Value of Bias 4 for Hiders per Generation
Value
Generation
0.7
0.6
0.5
0.4
0.3
0.2
0
100
200
300
400
500
600


Average Value of Bias 5 for Hiders per Generation
Value
Generation
0.0
0.2
0.4
0.6
0.8
0
100
200
300
400
500
600

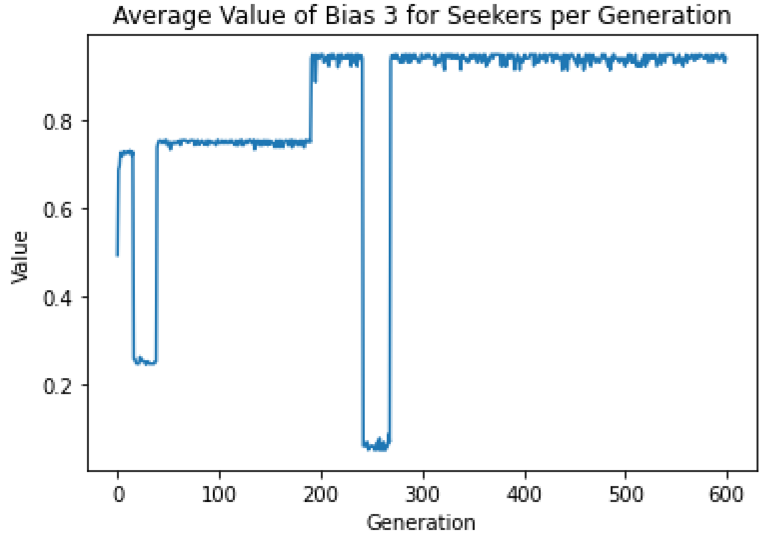

Average Value of Bias 0 for Seekers per Generation
Value
Generation
0.6
0.7
0.8
0.9
0
100
200
300
400
500
600


Average Value of Bias 1 for Seekers per Generation
Value
Generation
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
100
200
300
400
500
600

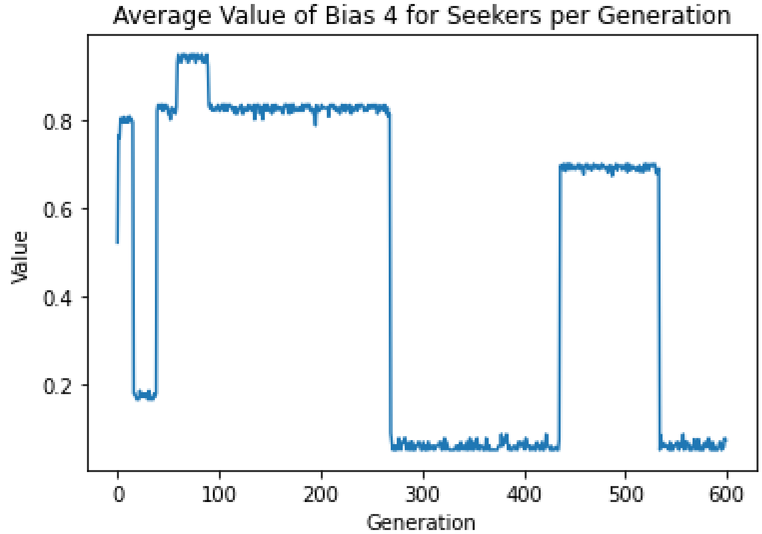
Average Value of Bias 2 for Seekers per Generation

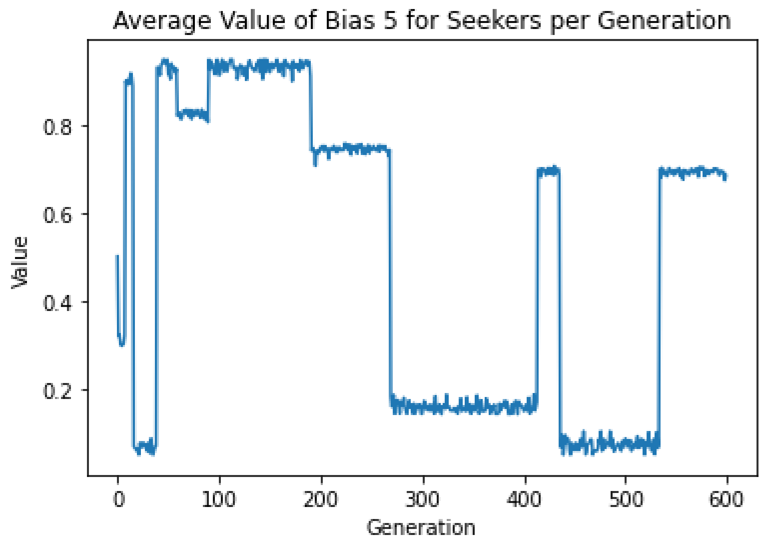


Average Value of Bias 3 for Seekers per Generation



Average Value of Bias 4 for Seekers per Generation





Future Work

In the final stages of my development, I was running up against the physical limits of my desktop. I believe that, if I were to continue development, I would greatly consider rewriting my code in a more productive language, like C++, seeing as I don't currently use any python machine learning libraries. Doing this would also allow me to make use of multiprocessing, which could speed up the generational process even more. This would allow me to observe longer periods of evolution across larger population sizes, which I believe would allow for more nuanced details to develop. I would also consider re-implementing environmental hazards; I had neglected most of them in order to re-add the hider intelligence and improve the algorithms I had already implemented, so seeing more detailed maps would offer a wider variety of opportunities for unique developments. I also see value in improving the fitness function, though the exact specifics of such currently escape me. It can be seen by the results of the third stage of testing that my seeker and hider fitnesses were beginning to plateau, despite the accuracy graph showing a large room for improvement.

Conclusions

After concluding the third stage of testing, I can conclude that the two intelligences are indeed learning and evolving in response to each other. As expected, the seeker takes massive stock in following the hider's scent trail, while all other biases change in accordance with the individual strategies at play. It appears, from what I assume to be the case, that the hider AIs are taking advantage of an oversight in my fitness function in order to give the seeker a lower fitness value due to "inaccurate" movement. Though the limitations of my hardware prevent me from observing an individual match, some interesting observations can be made by observing the bias evolutions. By the end of the 100th generation, the average hider intelligence had a less than average bias in favor of smell, a maximum bias in favor of moving away from the origin, a maximum bias in favor of active movement, an average bias in regards to the previous movement, no bias for time, and no bias in regard to passive movement. So it can be inferred that at this point the hider strategy was to move as far away from the origin as possible, then choosing to either continue in straight lines along the edge or double back on itself, always moving and never staying still. On the other hand, at the end of the 100th generation, the average seeker had a maximum bias in favor of smell, a below average bias in favor of moving away from the origin, no bias in favor of active movement, high bias in favor of previous movement, high bias in favor of time, and high bias in favor of passivity. It would appear then that the seeker strategy was to follow the hider's scent, typically following straight lines, becoming less random over time and more likely to follow the high probability directions. For some reason, there is high passivity, which I'm not quite sure why such is the case. It is probably the reason why these strategies combined result in around a 30% seeker win rate.

References

[1] Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2020, February 11). Emergent Tool Use From Multi-Agent Autocurricula. Date accessed 1/11/2021.

[2] Gunady, M., Gomaa, W., & Takeuchi, I. (2012, September 12). Multi-agent Task Division Learning in Hide-and-Seek Games. Date accessed 1/11/2021.

[3] Bloch, F., Dutta, B., & Dziubinski, M. (2020, January 09). A game of hide and seek in networks. Date accessed 1/11/2021.

[4] NRL, J., Trafton, J., Nrl, NRL, A., Schultz, A., NRL, D., . . . Authors: J. Gregory Trafton NRL. (2006, March 01). Children and robots learning to play hide and seek. Date accessed 1/11/2021.

[5] Weihs, L., Kembhavi, A., Han, W., Herrasti, A., Kolve, E., Schwenk, D., . . . Farhadi, A. (2019, December 18). Artificial Agents Learn Flexible Visual Representations by Playing a Hiding Game. Date accessed 1/11/2021.